

4

Uma descrição do domínio de SMAs usando ontologias e um exemplo de aplicação

Este capítulo apresenta uma descrição do domínio de SMAs através da definição de ontologias, assim como a aplicação destas ontologias na definição de um método para análise de modelos de SMAs descritos em linguagens de modelagem para SMAs.

4.1.

Preâmbulo

O uso de ontologias para descrição de domínios de conhecimento tem sido largamente difundido na comunidade de Ciência da Computação, notadamente em Inteligência Artificial, desde a última década do século XX, quando pesquisadores fizeram uma das primeiras referências ao termo ontologia fora da comunidade de filósofos (Neches et al, 1991). Mais recentemente, também a pesquisa voltada para a Engenharia de Software introduziu o uso de ontologias durante a fase de elicitação de requisitos, a fim de prover não só um melhor entendimento do domínio da aplicação a ser desenvolvida, como também descrevê-lo formalmente, permitindo a verificação de consistência entre o domínio da aplicação descrito na ontologia e os modelos conceituais definidos para a aplicação propriamente dita (Evermann & Wand, 2005; Dong et al., 2004; Kalfoglou & Robertson, 1999).

Por outro lado, a classe de aplicações desenvolvidas seguindo a abordagem multiagentes possui características específicas, as quais têm sido exploradas pela comunidade de Engenharia de Software a fim de facilitar o desenvolvimento de SMAs. Assim, vários resultados surgiram na pesquisa relacionada à definição de métodos e metodologias (Wooldridge et al, 2000; Wood & DeLoach, 2001), linguagens de modelagem (Bauer et al, 2001, Wagner, 2003; Silva & Lucena, 2004), construção de padrões de arquitetura (Sycara et al, 2003; FIPA, 2004) e produção de infra-estrutura básica de desenvolvimento

(Collis et al, 1998; Bellifemine et al, 1999; Bellifemine et al, 2003) para SMAs. Além disso, também foram empreendidos esforços na formalização das abstrações que fazem parte deste domínio (d’Inverno e Luck, 2001), com o intuito de prover suporte para verificação de SMAs. Neste contexto, e tendo como base a literatura sobre SMAs, foi definido o TAO (*Taming Agents and Objects*) (Silva et al, 2003) um arcabouço conceitual desenvolvido para proporcionar um melhor entendimento do domínio de SMAs e prover fundamentos para a engenharia deste tipo de sistemas. Em (Silva, 2004) foi feita uma extensão deste arcabouço a fim de descrever os aspectos dinâmicos de SMAs e o comportamento básico das entidades participantes de um SMA. Pode-se dizer que TAO e TAO estendido (doravante apenas TAO) estabelecem as entidades essenciais para o desenvolvimento de SMAs, assim como os relacionamentos entre estas entidades.

A formalização de um arcabouço conceitual para SMAs através da definição de uma ontologia que o descreva pode ser utilizada como suporte para a definição de métodos ou processos de análise, verificação ou validação de modelos ou de propriedades de SMAs. Neste capítulo serão apresentadas ontologias que descrevem o domínio de SMAs e uma aplicação destas ontologias como base para a definição de um método de estruturação e análise de modelos de SMAs descritos em linguagens de modelagem para SMAs. Este capítulo é estruturado da seguinte maneira: na seção 4.2 descreve-se o TAO, destacando-se as abstrações e os relacionamentos que ele especifica; na seção 4.3 apresenta-se o processo de desenvolvimento das ontologias que descrevem o domínio de SMAs baseada no TAO; na seção 4.4 apresenta-se uma aplicação da ontologia a partir da definição de um método para estruturação e análise de modelos de SMAs e, finalmente, na seção 4.5 apresenta-se uma discussão sobre o conteúdo do capítulo.

4.2.

O arcabouço conceitual TAO

O arcabouço conceitual TAO define uma ontologia que especifica os conceitos ou abstrações essenciais para o desenvolvimento de SMAs. Ao considerar o modelo de quatro camadas proposto pelo OMG (*Object Management Group*) e que descreve a arquitetura de metadados MOF (*Meta Object Facility*) (OMG, 2003), observa-se que o TAO encontra-se na camada referente aos metamodelos, pois as abstrações e relacionamentos que ele define são

candidatas a abstrações especificadas por linguagens de modelagem, métodos e metodologias que auxiliem no desenvolvimento de SMAs. As abstrações foram classificadas em três categorias: (i) abstrações fundamentais, (ii) abstrações de grupos, e (iii) abstrações de ambientes (Figura 14).

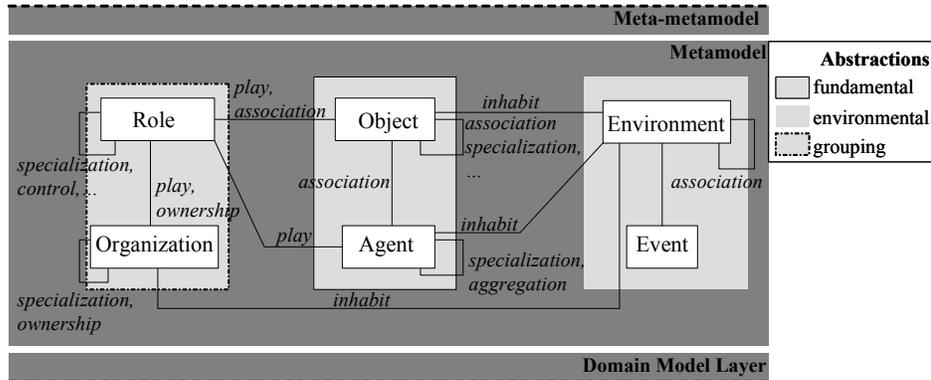


Figura 14. Abstrações e relacionamentos definidos no TAO (fonte (Silva et al, 2003)).

As abstrações fundamentais consideradas são os agentes e os objetos, pois considera-se que sejam entidades distintas no sentido de que um objeto não possui as propriedades necessárias para representar um agente. As abstrações de grupos representam a analogia sociológica que se faz ao considerar agentes inseridos no contexto de um sistema. Desta forma, agentes desempenham diferentes papéis em organizações que agrupam agentes que compartilham algum objetivo. Como toda organização está inserida num ambiente, de onde é possível perceber e disparar eventos, o TAO também considera como outra categoria as abstrações ambientais. A seguir faz-se uma descrição das entidades e relacionamentos especificados pelo TAO, e descritos nas ontologias como conceitos do domínio de SMAs.

4.2.1.

Entidades descritas no TAO

Objeto

Um objeto é uma entidade passiva ou reativa que possui estado e comportamento e que pode se relacionar a outras entidades. O estado de um objeto não possui estrutura pré-definida (Ishida et al, 1992) e é caracterizado pelas informações armazenadas nos seus atributos, que podem se referir a si mesmo, ao ambiente em que está inserido ou aos outros objetos. O comportamento de um objeto é dado pelas operações que ele pode realizar através da execução de métodos. Chamadas de métodos caracterizam a

interação entre objetos. Objetos participam de relacionamentos com outras entidades do sistema, como agentes, organizações e papéis. Uma classe de objeto define objetos com estrutura e comportamento similares e, portanto, diz-se que objetos são instâncias de classes de objetos.

Agente

Um agente é uma entidade autônoma, adaptativa e interativa, que possui estado mental. O estado mental de um agente é descrito através de suas crenças, metas, planos e ações e agrega o que se convencionou chamar de estado e comportamento de uma entidade. Assim, o estado do agente é descrito através de suas crenças e metas e o comportamento através dos planos e das ações definidas no agente. Planos são conjuntos de ações e estão associados às metas que eles atingem após serem executados. As crenças armazenam o conhecimento do agente relativo ao contexto em que ele está inserido e as metas são estados futuros que o agente deseja atingir. Agentes interagem através de protocolos de comunicação. Uma classe de agentes define agentes com estrutura e comportamento similares e, portanto, diz-se que agentes são instâncias de classes de agentes. Considerando-se a autonomia que caracteriza os agentes, o comportamento de agentes que são instâncias de uma mesma classe de agente pode diferir ao longo do ciclo de vida de cada agente, apesar de similares quando de suas instanciações.

Organização

Uma organização é uma entidade que agrupa agentes que desempenham papéis e possuem metas comuns. Organizações definem papéis que serão desempenhados por agentes e sub-organizações e restringem o comportamento destas entidades através de axiomas. O estado de uma organização estende o estado do agente acrescentando a ele os axiomas. O comportamento de uma organização é determinado pelo comportamento dos agentes e das sub-organizações que desempenham papéis na organização, eventualmente acrescidos de outros planos e ações. Apesar de não explicitado no TAO, infere-se que organizações que não desempenhem papéis se comunicam através de protocolos, os quais são compostos por seqüências de mensagens. Uma classe de organização define organizações com estrutura e comportamento similares e, portanto, diz-se que organizações são instâncias de classes de organizações.

Ambiente

Um ambiente é definido pelos seus habitantes, que podem ser agentes, objetos e organizações. Um ambiente pode ser ativo ou passivo e seu estado e

comportamento dependem das características determinadas pela passividade ou atividade do ambiente. Um ambiente passivo possui estado e comportamento similares aos objetos. Ambientes ativos possuem estado e comportamento similares aos agentes, porém não desempenham papéis em organizações. Apesar de não explicitado no TAO, infere-se que ambientes ativos se comunicam através de protocolos, os quais são compostos por seqüências de mensagens. Uma classe de ambiente define ambientes com estrutura e comportamento similares e, portanto, diz-se que ambientes são instâncias de classes de ambientes.

Papel de agente

Um papel de agente guia e restringe o comportamento do agente que o desempenha através da definição das metas que o agente deve atingir e das ações que ele poderá e deverá executar ao desempenhá-lo. O estado de um papel de agente é determinado por suas crenças e metas. O comportamento do papel de agente é determinado por seus direitos e deveres (ações que ele pode e deve executar, respectivamente) e pelos protocolos de interação com outros papéis de agente. Apesar de não explicitado no TAO, também é possível a interação entre as entidades que desempenham papel de agente com organizações e ambientes ativos, os quais não desempenham papéis porém conhecem protocolos de comunicação. Protocolos são compostos por mensagens que seguem uma ordem pré-estabelecida. Estas mensagens são trocadas entre os agentes ou sub-organizações que desempenham papéis com outros agentes ou sub-organizações que também desempenham papéis. Uma classe de papel de agente define papéis de agente com estrutura e comportamento similares e, portanto, diz-se que papéis de agente são instâncias de classes de papéis de agente.

Papel de objeto

Um papel de objeto guia e/ou restringe o comportamento de um objeto. Quando um objeto desempenha um papel, o acesso às informações que ele contém pode ser restringido pelo papel, ou ainda ele pode ter suas informações internas aumentadas, a partir das novas informações que o papel contém. O estado de um papel de objeto é similar ao estado do objeto, assim como o comportamento. Uma classe de papel de objeto define papéis de objeto com estrutura e comportamento similares e, portanto, diz-se que papéis de objeto são instâncias de classes de papéis de objeto.

4.2.2.

Relacionamentos descritos no TAO

Os relacionamentos descritos no TAO são *inhabit*, *play*, *ownership*, *control*, *dependency*, *association*, *aggregation* e *specialization/generalization*.

Inhabit

O relacionamento *inhabit* relaciona as entidades agentes, organizações ou objetos a ambientes, indicando que tais entidades estão definidas no contexto de determinado ambiente.

Play

O relacionamento *play* relaciona agentes e sub-organizações a papel de agente e objetos a papel de objetos, indicando que papel um agente, sub-organização ou objeto pode desempenhar ao longo de seu ciclo de vida.

Ownership

O relacionamento *ownership* relaciona organizações aos papéis de agente, que podem ser desempenhados por agentes ou por sub-organizações internamente a elas. Além deles, também relaciona organizações a papéis de objetos, que são desempenhados por objetos no contexto da organização.

Control

O relacionamento *control* relaciona papéis de agente, indicando que um papel de agente desempenhado por um agente ou sub-organização tem poderes de controle sobre outro papel de agente desempenhado por outro (ou eventualmente pelo mesmo) agente ou sub-organização.

Dependency

O relacionamento *dependency* está definido entre objetos; entre papéis de objeto; e entre papéis de agente e papéis (de agente ou objeto). Ele indica uma extensão do relacionamento de mesmo nome definido pelo metamodelo de UML (OMG, 2004).

Association

O relacionamento *association* está definido entre objetos e todas as outras entidades de SMAs, com exceção de ambientes; entre papéis de agente e papéis (de agentes e de objetos); entre papéis de objetos; e entre ambientes.

Este relacionamento indica que podem ocorrer interações entre as entidades relacionadas.

Aggregation

O relacionamento *aggregation* está definido entre objetos; entre papéis de objetos; e entre papéis de agentes. Este relacionamento indica que uma entidade (agregador) pode utilizar funcionalidades disponíveis na entidade relacionada (agregado).

Specialization

O relacionamento *specialization* está definido entre entidades de mesmo tipo. Este relacionamento é a extensão, para as entidades de SMAs, do relacionamento de mesmo nome definido pelo metamodelo de UML.

4.3.

Ontologias para o domínio de SMAs

O arcabouço conceitual TAO define uma ontologia para o domínio de SMAs descrita em linguagem natural. Sua descrição formal permite não só o reuso das informações deste domínio em aplicações diversas e relacionadas à utilização da abordagem multiagente, como também a avaliação da consistência do TAO enquanto ontologia para SMAs.

Considerando que o TAO foi definido a partir de pesquisa exaustiva sobre a literatura relativa ao desenvolvimento de SMAs, e que o mesmo aborda entidades e relacionamentos que não necessariamente são considerados em algumas abordagens para desenvolvimento de SMAs, optou-se pela descrição formal do TAO em duas fases, cada uma delas representada por uma ontologia. Assim, na primeira fase é definida uma ontologia mais flexível, que não contempla restrições absolutas às entidades e relacionamentos especificados no TAO, e na segunda fase estende-se a primeira ontologia acrescentando-lhe restrições aos conceitos e relacionamentos nela especificados.

A primeira ontologia é composta apenas por conceitos associados às entidades e aos relacionamentos definidos no TAO, acrescidos das restrições hierárquicas e das disjunções entre tais conceitos. Nesta ontologia também são definidas propriedades que podem ser usadas para definir axiomas que restrinjam ainda mais os conceitos especificados ou para definir consultas relacionadas às instâncias destes conceitos. A segunda ontologia estende a primeira a partir da introdução de axiomas que restringem os conceitos e as

propriedades já definidos, com o intuito de definir, ainda que parcialmente, a estrutura das entidades e dos relacionamentos especificados como conceitos na primeira ontologia.

Para o desenvolvimento formal das ontologias foi adotada uma abordagem do tipo *top-down* (Gómez-Pérez, 1999). Nesta abordagem parte-se de conceitos mais abstratos e relacionados ao domínio descrito os quais são refinados ao longo do desenvolvimento da ontologia. A linguagem escolhida para descrição das ontologias foi a lógica de descrição (DL) ALCHIQr+(D)- (2.3.1). As ontologias definidas nas duas fases de descrição formal do TAO compartilham os mesmos conceitos e propriedades, logo as subseções 4.3.1 e 4.3.2 referem-se a ambas ontologias. Na subseção 4.3.3 são descritas algumas das propriedades consideradas como relativas ao domínio de SMAs e suas respectivas traduções como axiomas introduzidos na ontologia estendida, definida na segunda fase da descrição formal do TAO. Tais propriedades são de interesse durante a análise de modelos que descrevem SMAs.

4.3.1.

Conceitos das ontologias

A definição do TAO foi feita a partir de abordagem similar aos modelos de entidade e relacionamento (ER). Assim, definiu-se os conceitos abstratos `entity` e `relationship` como conceitos iniciais da ontologia, a serem refinados. O conceito `entity` representa as entidades que fazem parte de um SMA e o conceito `relationship` representa os relacionamentos definidos entre estas entidades.

A partir da descrição das entidades especificadas no TAO e descritas na subseção 4.2.1, iniciou-se o refinamento do conceito `entity`. Assim, foram definidos os conceitos `class` e `class-instance`, para representar os tipos diferentes de entidades, como sub-conceitos⁴ do conceito `entity`. A partir deste primeiro refinamento, convencionou-se que os conceitos referentes a entidades do tipo classe seriam todos definidos como sub-conceitos do conceito `class` e receberiam nome com sufixo `-class`. Conceitos referentes a entidades do tipo

⁴ Um conceito `A` é sub-conceito do conceito `B` se todo indivíduo do conceito `A` é indivíduo do conceito `B`.

instância de classe seriam todos definidos como sub-conceitos do conceito `class-instance` e receberiam o mesmo nome, traduzido para o inglês.

Os conceitos `class` e `class-instance` serviram como base efetiva para a especificação das entidades descritas no TAO. Iniciando-se a descrição dos conceitos da ontologia a partir dos sub-conceitos de `class`, foram criados os conceitos abstratos `citizen-class`, `role-class` e `environment-class`. Os indivíduos do tipo `citizen-class` englobam as classes de entidades que possuem capacidade de habitar um ambiente, ou seja, classes de agente (`agent-class`), classes de organizações (`organization-class`) e classes de objetos (`object-class`), todos eles definidos como sub-conceitos de `citizen-class`. Os indivíduos do tipo `role-class` englobam as classes de papéis de agente (`agent-role-class`) e de objeto (`object-role-class`). Os indivíduos do tipo `environment-class` englobam as classes de ambientes ativos (`active-environment-class`) e passivos (`passive-environment-class`). Para descrever indivíduos que representem as sub-organizações decidiu-se pelo refinamento do conceito `organization-class`, dividindo-o em dois sub-conceitos disjuntos: `main-organization-class` e `sub-organization-class`. Neste caso, apenas indivíduos representados pelo conceito `sub-organization-class` são habilitados a desempenhar papéis no contexto de outras organizações. A descrição hierárquica e consequente refinamento do conceito `class` é apresentado na Figura 15.

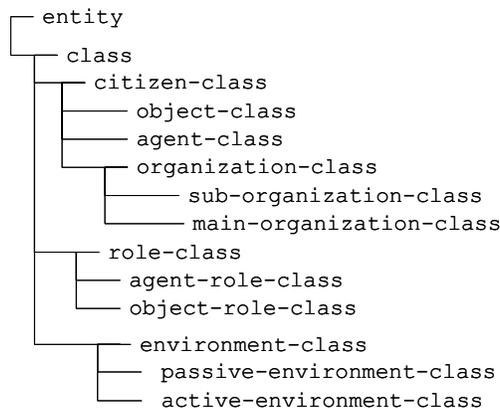


Figura 15 Refinamento do conceito `class`.

O refinamento do conceito `class-instance` foi mais direto, simplesmente definindo cada entidade representativa de uma instância de classe como um sub-conceito de `class-instance`. Assim, foram definidos os conceitos `agent` representando instâncias do conceito `agent-class`;

organization representando instâncias do conceito organization-class; main-organization representando instâncias do conceito main-organization-class; sub-organization representando instâncias do conceito sub-organization-class; object representando instâncias do conceito object-class; agent-role representando instâncias do conceito agent-role-class; object-role representando instâncias do conceito object-role-class; p-environment representando instâncias do conceito passive-environment-class; e a-environment representando instâncias do conceito active-environment-class. A descrição hierárquica e consequente refinamento do conceito class-instance é apresentado na Figura 16.

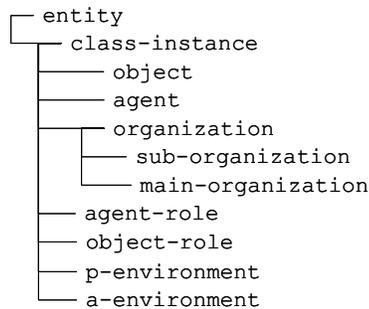


Figura 16 Refinamento do conceito class-instance.

O refinamento do conceito relationship se deu diretamente com a definição de conceitos cujos nomes coincidem com os relacionamentos previstos no TAO (subseção 4.2.2), a saber: inhabit, play, ownership, control, association, specialization e aggregation. Relacionamentos foram descritos como conceitos da ontologia seguindo a correspondência entre modelos de entidade/relacionamento (ER) e lógica de descrição (DL) descrita em (Baader et al, 2003). A descrição hierárquica e consequente refinamento do conceito relationship é apresentado na Figura 17.

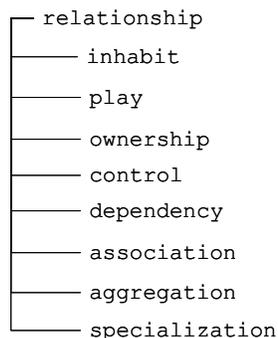


Figura 17 Refinamento do conceito relationship.

A fim de descrever a estrutura dos indivíduos do tipo `class`, foi definido o conceito `features`, o qual engloba conceitos que possibilitam a descrição da estrutura interna das entidades através da utilização de propriedades da ontologia, e não apenas como sub-conceito de outros conceitos. O conceito `goal` foi definido para representar as metas dos agentes, das organizações, dos papéis de agente e dos ambientes ativos. O conceito `belief` foi definido para representar as crenças que os agentes, as organizações, os papéis de agente e os ambientes ativos possuem. O conceito `axiom` foi definido para representar as regras que as organizações impõem para as entidades que dela participam. O conceito `plan` foi definido para representar os planos que um agente, uma organização ou um ambiente ativo podem desempenhar ao longo de seus ciclos de vida. O conceito `action` foi definido para representar os componentes atômicos de um plano, assim como para representar as ações conhecidas por agentes, organizações e ambientes ativos. Os conceitos `duty` e `right` foram definidos para representar as ações que agentes devem e podem executar quando estiverem desempenhando um determinado papel de agente. O conceito `protocol` foi definido para representar protocolos de interação entre agentes. Protocolos, por sua vez, são definidos por sequências de mensagens. O conceito `msg` foi definido para representar tais mensagens. Para representar o estado e comportamento de objetos foram definidos os conceitos `attribute` e `method`. Além disso, para diferenciar as mensagens trocadas entre objetos das mensagens trocadas entre agentes, organizações e ambientes ativos, o conceito `msg` foi refinado, sendo definidos os sub-conceitos `agent-msg` e `object-msg`. A descrição hierárquica e consequente refinamento do conceito `features` é apresentado na Figura 18.



Figura 18 Refinamento do conceito `features`.

Para descrever melhor os conceitos da ontologia foram definidas várias propriedades usadas com o intuito de relacionar ou restringir seus conceitos.

4.3.2.

Propriedades das ontologias

Uma ontologia é composta de conceitos, propriedades e axiomas. Numa ontologia, uma propriedade é representada por um relacionamento binário e direcionado entre conceitos. Este relacionamento pode representar uma função, pode satisfazer a propriedade transitiva, ou ainda possuir um relacionamento inverso. Nas ontologias que representam o domínio de SMAs descrito no TAO, definiu-se várias propriedades, que são listadas a seguir.

A propriedade `is-instanceOf` é usada para descrever o fato de que indivíduos do tipo `class-instance` são instâncias de indivíduos do tipo `class`. Esta propriedade é herdada pelos sub-conceitos de `class-instance` e `class`.

```
... :roles(...
(is-instanceOf :domain class-instance
               :range class)...
```

Figura 19 Propriedade `is-instanceOf`.

A Figura 19 mostra a definição formal da propriedade na ontologia usando a sintaxe implementada pelo sistema RACER. Esta definição é equivalente aos axiomas apresentados na Figura 20. O uso de axiomas para descrever as demais propriedades da ontologia só é feito quando o domínio ou a imagem das mesmas não são definidos como na Figura 19 (assinatura da TBox - 2.3.1).

```
(implies *top* all is-instanceOf class)
(implies some is-instanceOf *top* class-instance)
```

Figura 20 Axiomas equivalentes à definição de `is-instanceOf`.

Os relacionamentos especificados no TAO foram descritos como conceitos das ontologias pois eles derivam de um modelo ER e, de acordo com (Baader et al, 2003) em seu capítulo 4, relacionamentos devem ser reificados quando traduzidos para DL. Neste caso, instâncias dos relacionamentos representam as tuplas que os definem. Para melhor descrever os conceitos que representam relacionamentos nas ontologias, foram definidas duas propriedades: `has-end1` e `has-end2`.

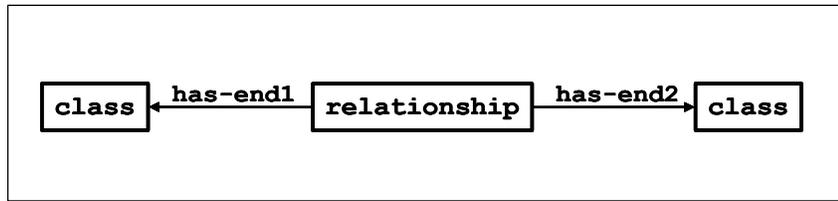


Figura 21. Estrutura dos relacionamentos definidos na ontologia.

Nestas propriedades, o domínio é definido por instâncias do conceito `relationship` e a imagem é definida como instâncias do conceito `class`. Estas propriedades permitem identificar os extremos do relacionamento considerado (Figura 21). Também foram definidas as propriedades inversas a estas: `is-end1` e `is-end2`, respectivamente (Figura 22).

```

... :roles(...)
(has-end1 :domain relationship
          :range class
          :inverse is-end1)
(has-end2 :domain relationship
          :range class
          :inverse is-end2)
(is-end1 :domain class
         :range relationship
         :inverse has-end1)
(is-end2 :domain class
         :range relationship
         :inverse has-end2) ...

```

Figura 22. Propriedades `has-end*` e `is-end*`.

Outras propriedades foram definidas para melhor descrever a estrutura interna das entidades. Assim, considerando-se que agentes, organizações e ambientes ativos têm metas e crenças e executam planos e/ou ações, etc. definiu-se as propriedades `has-goal`, `has-belief`, `has-plan` e `has-action`. Analogamente, definiu-se a propriedade `has-axiom` para, em conjunto com as propriedades previamente definidas para descrever agentes, permitir a descrição das organizações. Para descrever os papéis de agentes também foram definidas as propriedades `has-duty`, `has-right` e `has-protocol` (Figura 23). Cabe observar que a maioria das propriedades não teve seu domínio definido, pois isto é feito a partir da definição de axiomas introduzidos na segunda ontologia. Para descrever as propriedades internas de objetos, papéis de objetos e ambientes passivos definiu-se as propriedades `has-attribute` e `has-method`.

```

...
:roles (...)
  (has-goal :range goal)

```

```
(has-belief :range belief)
(has-action :range action)
(has-plan :range plan)
(has-axiom :range axiom)
(has-duty :domain agent-role-class
          :range duty)
(has-right :domain agent-role-class
          :range right)
(has-protocol :range protocol)
(has-attribute :range attribute)
(has-method :range method) ...)
```

Figura 23. Propriedades adicionais para definição de restrições a conceitos.

Além da definição das restrições aos conceitos da ontologia, as propriedades também podem ser utilizadas durante eventuais consultas a dados resultantes da instanciação da ontologia.

Os conceitos e as propriedades das ontologias descritos até o momento, acrescido de axiomas que especificam a hierarquia e a disjunção (Figura 24) entre os conceitos, definem a primeira ontologia. Assim, a primeira ontologia descreve formalmente as entidades e os relacionamentos descritos no TAO, sem especificar propriedades que restringem ou explicam estas entidades e relacionamentos. A segunda ontologia estende a primeira com a adição de axiomas que descrevem as propriedades das entidades e relacionamentos que, de acordo com o TAO, devem existir numa SMA. Estas propriedades apresentam restrições relativas às entidades e aos relacionamentos previstos no TAO.

```
(implies citizen-class class)
(implies role-class class)
(implies environment-class class)
(implies agent-class citizen-class)
...
(disjoint citizen-class role-class environment-class)
(disjoint agent-class organization-class object-class)
...
```

Figura 24. Hierarquia e disjunção entre conceitos.

4.3.3.

Restringindo conceitos: propriedades de SMAs e axiomas da ontologia

A fim de descrever as propriedades internas das entidades e dos relacionamentos especificados como conceitos na primeira ontologia foram definidos axiomas de inclusão. Estes axiomas utilizam as propriedades da

ontologia previamente definidas (4.3.2) e que caracterizam as entidades. Como exemplo de propriedades internas apresenta-se aquelas descritas no TAO para caracterizar agentes. A descrição será feita em duas partes, a primeira descrevendo propriedades relacionadas ao estado e comportamento do agente, e a segunda relacionada as outras entidades que se relacionam com os agentes. A descrição de casos gerais é apresentada quando pertinente. As propriedades SMA 1, SMA 2 e SMA 3 referem-se ao estado interno do agente.

Propriedade SMA 1. *Todo agente possui pelo menos uma meta.*

A Propriedade SMA 1 indica que indivíduos descritos pelo conceito `agent-class` devem estar associados a algum indivíduo definido pelo conceito `goal`. Como esta associação é feita através de `has-goal`, o axioma que descreve tal propriedade é o apresentado na Figura 25 .

```
(implies agent-class (some has-goal goal))
```

Figura 25. Axioma que restringe o conceito `agent-class` I.

Propriedade SMA 2. *Um agente possui crenças.*

A Propriedade SMA 2 indica que indivíduos descritos pelo conceito `agent-class` devem estar associados a pelo menos um indivíduo definido pelo conceito `belief`. Como esta associação é feita através de `has-belief`, o axioma que descreve tal propriedade é o apresentado na Figura 26.

```
(implies agent-class (some has-belief belief))
```

Figura 26. Axioma que restringe o conceito `agent-class` II.

Propriedade SMA 3. *Um agente possui planos.*

A Propriedade SMA 3 indica que indivíduos descritos pelo conceito `agent-class` devem estar associados a pelo menos um indivíduo definido pelo conceito `plan`. Como esta associação é feita através de `has-plan`, o axioma que descreve tal propriedade é o apresentado na Figura 27.

```
(implies agent-class (some has-plan plan))
```

Figura 27. Axioma que restringe o conceito `agent-class` III.

Como fórmula geral, uma entidade representada pelo conceito `x-class` tem seu estado e comportamento descrito pelas partes representadas por sub-conceitos do conceito `features`, aqui chamados `y`, é parcialmente descrita como apresentado na Figura 28.

```
(implies x-class (some has-y y))
```

Figura 28. Descrição geral de propriedade interna da entidade x-class.

As propriedades SMA 4 e SMA 5 referem-se às restrições que os agentes apresentam relativamente aos relacionamentos que os mesmos mantêm com outras entidades do domínio de SMAs.

Propriedade SMA 4. *Um agente deve desempenhar pelo menos um papel.*

A Propriedade SMA 4 indica que indivíduos descritos pelo conceito `agent-class` devem estar relacionados a indivíduos do conceito `agent-role-class` através de indivíduos do conceito `play` e das propriedades `has-end1` e `has-end2` e, eventualmente, suas inversas `is-end1` e `is-end2`, respectivamente. O axioma que descreve tal propriedade é apresentado na Figura 29.

```
(implies agent-class
  (some is-end1 (and play (all has-end2 agent-role-class))))
```

Figura 29. Axioma que restringe o conceito `agent-class` IV.

Propriedade SMA 5. *Um agente deve habitar um ambiente.*

A Propriedade SMA 5 indica que indivíduos descritos pelo conceito `agent-class` devem estar relacionados a indivíduos do conceito `environment-class` através de indivíduos do conceito `inhabit` e das propriedades `has-end1` e `has-end2` e, eventualmente, suas inversas `is-end1` e `is-end2`, respectivamente. O axioma que descreve tal propriedade é apresentado na Figura 30.

```
(implies agent-class (some is-end1
  (and inhabit (all has-end2 environment-class))))
```

Figura 30. Axioma que restringe o conceito `agent-class` V.

Como fórmula geral, uma entidade representada pelo conceito `x-class` que pode manter o relacionamento representado pelo conceito `xy-rel`, sub-conceito de `relationship`, com a entidade representada pelo conceito `y-class` tem sua descrição apresentada no axioma descrito na Figura 31.

```
(implies x-class
  (some is-end1 (and xy-rel (all has-end2 y-class))))
```

Figura 31. Descrição geral dos relacionamentos dos quais x-class faz parte.

Os relacionamentos descritos no TAO também especificam propriedades relativas ao domínio de SMAs. Estas propriedades referem-se às entidades que

podem participar de cada relacionamento. A Propriedade SMA 6 é um exemplo deste tipo de propriedade, a qual descreve o relacionamento `inhabit`.

Propriedade SMA 6. *O relacionamento `inhabit` deve relacionar indivíduos do conceito `citizen-class` a indivíduos do conceito `environment-class`.*

A Propriedade SMA 6 especifica que indivíduos pertencentes ao conceito `inhabit` são indivíduos que só estão associados a indivíduos pertencentes aos conceitos `citizen-class` e `environment-class`, através das propriedades `has-end1` e `has-end2`, respectivamente, como mostrado na **Figura 32**.

```
(implies inhabit
  (and (all has-end1 citizen-class)
        (all has-end2 environment-class)))
```

Figura 32. Formalização do relacionamento `inhabit`.

Outro exemplo de propriedade referente à estrutura de relacionamento é a Propriedade SMA 7, a qual descreve o conceito que define o relacionamento `play`.

Propriedade SMA 7. *O relacionamento `play` deve relacionar indivíduos dos conceitos `agent-class` ou `sub-organization-class` a indivíduos do conceito `agent-role-class`; ou ainda indivíduos do conceito `object-class` a indivíduos do conceito `object-role-class`.*

A **Figura 33** mostra a especificação da Propriedade SMA 7, classificando o relacionamento `play` como o conjunto de indivíduos que associam indivíduos do conceito `agent-class` ou `sub-organization-class` a indivíduos do conceito `agent-role-class`; ou indivíduos do conceito `object-class` a indivíduos do conceito `object-role-class`.

```
(implies play
  (or (and (all has-end1
            (or agent-class sub-organization-class))
          (all has-end2 agent-role-class))
      (and (all has-end1 object-class)
          (all has-end2 object-role-class))))
```

Figura 33. Formalização do relacionamento `play`.

De modo geral, a descrição de um conceito que define um relacionamento previsto no TAO pode ser feita através da união de intersecções de conceitos definidos a partir de axiomas, como é o caso do relacionamento `play` (**Figura 33**), ou pela simples intersecção de conceitos definidos a partir de axiomas, como é o caso do relacionamento `inhabit` (**Figura 32**). A fórmula geral para representar os conceitos que descrevem relacionamentos é

apresentada na Figura 34, onde `rel` representa indivíduos pertencentes ao conceito, e `N`, ($N \geq 1$) indica o número de uniões necessárias para descrever os possíveis pares de indivíduos que participam do relacionamento `rel`.

```
(implies rel (or
  (and (all has-end1 x1-class)
        (all has-end2 y1-class))...
  (and (all has-end1 xN-class)
        (all has-end2 yN-class))))
```

Figura 34. Descrição geral dos relacionamentos.

4.4.

Aplicação das ontologias: um método para estruturação e análise de modelos de SMAs

Durante a definição do TAO observou-se que nele eram especificadas abstrações que poderiam ser utilizadas durante a definição de linguagens de modelagem, métodos e metodologias que auxiliassem o desenvolvimento de SMAs. Neste sentido, o TAO foi usado como referência para a definição do metamodelo que originou a linguagem de modelagem MAS-ML (Silva, 2004), a definição de um método arquitetural orientado a aspectos e de uma linguagem de padrões associados, por exemplo, às propriedades de autonomia e mobilidade de agentes (Garcia, 2004). Assim, uma das aplicações possíveis para sua formalização é a análise de modelos descritos na linguagem de modelagem que ele originou. Partindo deste pressuposto e considerando que o TAO descreve o domínio de SMAs, definiu-se um método para estruturação e análise de modelos de SMAs descritos em linguagens de modelagem orientadas a agentes. O método, chamado Observed-MAS (*Ontology-Based method for Structuring and analysing the Design of Multi-Agent Systems*), é apresentado brevemente nas subseções seguintes. No Capítulo 5 o método é aplicado para a linguagem de modelagem MAS-ML e no Capítulo 7 é apresentada uma aplicação para a linguagem AUML (Bauer et al, 2001).

4.4.1.

Visão geral do método

O método Observed-MAS dá suporte à estruturação de modelos de SMAs, descritos em linguagens de modelagem, através do uso de ontologias. Estas ontologias descrevem os termos e as propriedades do domínio de SMAs, assim

como os termos e as propriedades da linguagem de modelagem na qual o sistema foi modelado, o que permite verificar a boa formação da estruturação dos modelos. Além do suporte à estruturação, o método também dá suporte à análise dos modelos através de consultas à base de conhecimento, a qual é gerada a partir da representação dos modelos como instâncias da ontologia.

O Observed-MAS é composto por duas fases distintas, cada uma delas contendo atividades relacionadas à estruturação e à análise dos modelos de SMAs. A FASE UM (F1) é voltada para a estruturação e análise individual de cada diagrama que compõe a modelagem do sistema, considerando-se o domínio de SMAs e as propriedades estruturais de cada diagrama, previstas no metamodelo da linguagem de modelagem adotada. A FASE DOIS (F2) é voltada para a estruturação e análise do conjunto de diagramas que compõem a modelagem do sistema, com foco nas propriedades de interdependência entre diagramas, a fim de gerar uma modelagem final que seja consistente com essas propriedades. A Figura 35 ilustra uma visão geral do método.

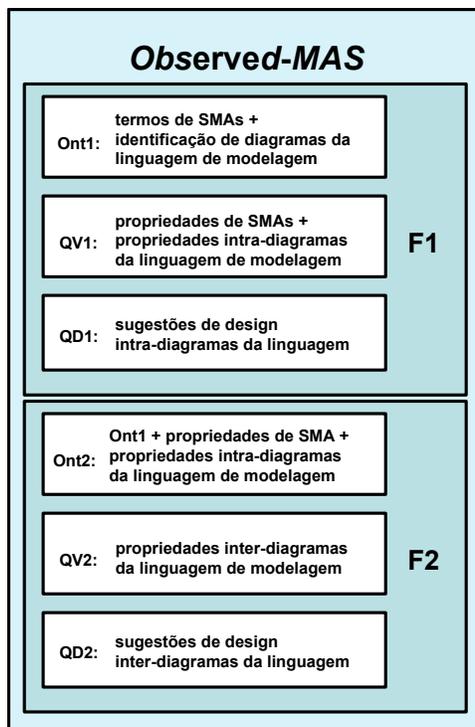


Figura 35. Visão geral do método Observed-MAS.

A estruturação dos modelos feita pelo Observed-MAS é apoiada por ontologias (Ont1 e Ont2) que descrevem o domínio de SMAs e o metamodelo que especifica a linguagem de modelagem. Assim, diagramas da linguagem de modelagem que modelam o sistema são diretamente mapeados para instâncias das ontologias, que ficam armazenadas em bases de conhecimento. A análise

dos modelos feita pelo Observed-MAS é apoiada por consultas feitas à base de conhecimento, através de sistemas que possuem o aparato dedutivo necessário para manipulação deste tipo de dados.

As ontologias Ont1 e Ont2 são desenvolvidas a partir das ontologias usadas na formalização do TAO, acrescidas de novos conceitos e propriedades que descrevem as especificidades da linguagem de modelagem adotada.

O método define, em cada fase, dois conjuntos de consultas para as atividades de análise: um destinado a encontrar violações nas propriedades internas definidas para cada diagrama (QV1), ou nas propriedades de interdependência entre eles (QV2); e outro composto por consultas que sugerem ao desenvolvedor boas práticas de modelagem (QD1 e QD2). Estas consultas são definidas a partir da natureza estática das ontologias utilizadas e do conhecimento *a priori* das propriedades inerentes aos diagramas.

4.4.2.

Fase F1: Estruturação e análise das propriedades de SMAs e propriedades intra-diagramas

A fase F1 do método Observed-MAS subdivide-se em três etapas distintas, cada uma delas com uma finalidade específica. A primeira etapa refere-se à atividade de estruturação dos diagramas da fase F1, e consiste da primeira ontologia (Ont1), cujos conceitos são definidos pelos termos que compõem o vocabulário relacionado ao domínio de SMAs (primeira ontologia do TAO) e ao metamodelo da linguagem de modelagem. Restrições que descrevem a hierarquia entre os conceitos também fazem parte de Ont1, assim como restrições de disjunção entre conceitos.

A Ont1 foi definida considerando-se poucas restrições entre seus conceitos, a fim de dar liberdade ao desenvolvedor durante as primeiras atividades de modelagem, pois é natural o aparecimento de inconsistências nos modelos durante a criação dos diagramas. Assim garante-se que instâncias de Ont1, produzidas a partir de diagramas incompletos, não geram bases de conhecimento inconsistentes.

Além destas restrições, a Ont1 também contém a definição de várias propriedades que relacionam seus conceitos. Estas propriedades têm papel importante nas atividades de análise da fase F1, pois são utilizadas pelas consultas à base de conhecimento para descobrir violações as propriedades intra-diagramas da linguagem de modelagem. Propriedades intra-diagramas são

aquelas relacionadas a especificação dos diagramas definida pelo metamodelo da linguagem de modelagem.

A segunda etapa da fase F1 é composta por conjuntos de consultas previamente definidas, um deles contendo consultas de âmbito geral, referentes ao domínio de SMAs, e os demais referentes a consultas no contexto de cada diagrama previsto pela linguagem de modelagem, sejam estes diagramas estáticos ou dinâmicos. A união destes conjuntos é descrita na Figura 35 como QV1. Estas consultas analisam violações as propriedades de SMAs descritas na Seção 4.3.3 e as propriedades estruturais dos diagramas descritos na linguagem de modelagem adotada. A consulta ilustrada na Figura 36 retorna como resposta indivíduos que violam a Propriedade SMA 1.

```
(retrieve (?agentwithoutgoal)
  (and (?agentwithoutgoal agent-class)
    (?agentwithoutgoal NIL has-goal)))
```

Figura 36. Consulta sobre violação à Propriedade SMA 1.

Consultas relacionadas à estrutura interna das entidades seguem o mesmo padrão da consulta apresentada na Figura 36. Assim, para *x-class* uma classe que representa um sub-conceito de *class* e *y* um sub-conceito de *features*, a Figura 37 apresenta a fórmula geral para consultas deste tipo.

```
(retrieve (?entity)
  (and (?entity x-class)
    (?entity NIL has-y)))
```

Figura 37. Fórmula geral: consulta sobre estrutura interna da entidade.

Outro exemplo de consulta sobre violações a propriedades de SMAs é ilustrado na Figura 38. Esta consulta retorna como resposta indivíduos que violam a Propriedade SMA 4.

```
(retrieve (?agentwithoutrole)
  (and (?agentwithoutrole agent-class)
    (not (?agentwithoutrole
      (some is-end1
        (and play (some has-end2 agent-role-class)))))))
```

Figura 38. Consulta sobre violação a propriedade **Propriedade SMA 4**.

Consultas relacionadas aos relacionamentos que cada entidade está habilitada a participar seguem o mesmo padrão da consulta apresentada na Figura 38. Assim, para *x-class* e *y-class* sub-conceitos de *class* e *xy-rel* sub-conceito de *relationship*, a Figura 39 apresenta a fórmula geral para consultas deste tipo.

```
(retrieve (?entity)
  (and (?entity x-class)
    (not (?entity (some is-end1 (and xy-rel
      (some has-end2 y-class)))))))
```

Figura 39. Fórmula geral: consulta a relacionamentos entre x-class e y-class.

Quando consideradas as consultas a violações estruturais na definição dos relacionamentos pode-se defini-las de duas formas: através da consulta ao conjunto explícito de indivíduos que é complementar ao habilitado a participar do relacionamento (Figura 40); ou através da consulta que considera a negação do conjunto de indivíduos habilitados a participar do relacionamento (complementar implícito - Figura 42).

A consulta ilustrada na Figura 40 analisa a base de conhecimento e retorna como resposta violações a Propriedade SMA 6. Ela foi definida a partir do complementar dos conjuntos de classes que definem as restrições ao domínio e a imagem das propriedades `has-end1` e `has-end2`.

```
(retrieve (?end1 ?end2 ?inh)
  (and (?inh inhabit)
    (?inh ?end1 has-end1)
    (?inh ?end2 has-end2)
    (or (?end1 (or environment-class role-class))
      (?end2 (or citizen-class role-class))))
```

Figura 40. Consulta sobre violação à Propriedade SMA 6.

A fórmula geral para definir consultas sobre relacionamentos mal estruturados usando o complementar explícito é apresentada na Figura 41. Neste caso, `xy-rel` é sub-conceito de `relationship` e representa o relacionamento considerado, `x1-class ... xN-class` representam os sub-conceitos de `class` que não estão habilitados para serem `end1` de `xy-rel` e `y1-class ... yK-class` representam os sub-conceitos de `class` que não estão habilitados para serem `end2` de `xy-rel`.

```
(retrieve (?end1 ?end2 ?xyrel)
  (and (?xyrel xy-rel)
    (?xyrel ?end1 has-end1)
    (?xyrel ?end2 has-end2)
    (or (?end1 (or x1-class x2-class ... xN-class))
      (?end2 (or y1-class y2-class ... yK-class))))
```

Figura 41. Fórmula geral: consulta sobre violação à estrutura de relacionamentos I.

A consulta ilustrada na Figura 42 foi definida a partir da negação da Propriedade SMA 7 e retorna como resposta relacionamentos `play` mal estruturados.

```
(retrieve ($?end1 $?end2 ?play) (and
```

```
(?play play)
($?end1 ?play is-end1)
($?end2 ?play is-end2)
(not (and
      (?end1 (or agent-class sub-organization-class))
      (?end2 agent-role-class)))
(not (and ($?end1 object-class)
          ($?end2 object-role-class))))
```

Figura 42. Consulta sobre violação à Propriedade SMA 7.

A fórmula geral para definir consultas sobre relacionamentos mal estruturados usando o negação de propriedade é apresentada na Figura 43. Neste caso, *xy-rel* é sub-conceito de *relationship* e representa o relacionamento considerado e, para $i \geq 1$, *x_{i1}-class ... x_{iN_i}-class* representam os sub-conceitos de *class* que estão habilitados para serem *end1* de *xy-rel* e *y_{i1}-class ... y_{iK_i}-class* representam os sub-conceitos de *class* que estão habilitados para serem *end2* de *xy-rel*. Observa-se que a consulta usando negação pode ser bem mais complexa que a consulta usando complementar explícito.

```
(retrieve ($?end1 $?end2 ?xyrel) (and
  (?xyrel xy-rel)
  ($?end1 ?xyrel is-end1)
  ($?end2 ?xyrel is-end2)
  (not (and
        (?end1 (or x11-class x12-class ... x1N1-class))
        (?end2 (or y11-class y12-class ... y1K1-class))))
    ...
  (not ($?end1 (or xM1-class xM2-class ... xMNm-class))
        ($?end2 (or yM1-class yM2-class ... yMKm-class))))
```

Figura 43. Fórmula geral: consulta sobre violação à estrutura de relacionamentos II.

A terceira etapa é composta por um conjunto de consultas (QD1) que retornam como resposta sugestões de boas práticas de modelagem de SMAs. Neste caso as consultas são sempre dependentes da forma como a linguagem de modelagem adotada especifica determinados aspectos de SMAs, como por exemplo, a comunicação através de protocolos. É esperado que esta etapa seja enriquecida ao longo do tempo, com a definição de novas consultas, a partir do amadurecimento das metodologias e das linguagens de modelagem para SMAs.

4.4.3.

Fase F2: Estruturação e análise das propriedades de interdependência entre diagramas

A fase F2 do Observed-MAS (Figura 35) também é composta de três etapas distintas. A primeira etapa refere-se à atividade de estruturação do conjunto de diagramas identificados na fase F1. Esta etapa especifica a segunda ontologia (Ont2), que é uma extensão conservativa de Ont1, obtida a partir da adição de axiomas que definem a semântica das entidades e dos relacionamentos previstos no domínio de SMAs, assim como axiomas que descrevem a estrutura dos diagramas especificados pelo metamodelo da linguagem de modelagem. Alguns dos axiomas foram definidos durante a descrição formal das propriedades de SMAs (4.3.3). Observa-se que estes axiomas refletem as consultas QV1 definidas na fase F1 (4.4.2).

As mesmas propriedades definidas em Ont1, e usadas na fase F1, também têm papel importante tanto nas atividades de estruturação quanto nas atividades de análise da fase F2. Nas atividades de estruturação as propriedades da ontologia são usadas na descrição da semântica dos conceitos que representam as entidades e relacionamentos de SMAs e do metamodelo da linguagem de modelagem. Nas atividades de análise, as propriedades da ontologia são utilizadas pelas consultas a base de conhecimento para descobrir violações nas propriedades inter-diagramas da linguagem de modelagem. Propriedades inter-diagramas são aquelas relacionadas a especificação das interdependências entre diagramas, definidas pelo metamodelo da linguagem de modelagem.

A segunda etapa da fase F2 é composta por um conjunto de consultas (QV2) que analisam a base de conhecimento, gerada pelos diagramas que compõem a modelagem de um SMA, e retornam como resposta informações sobre violações as propriedades de interdependência entre diagramas, previstas pela especificação do metamodelo da linguagem. Dadas suas peculiaridades, esta etapa é totalmente dependente da linguagem de modelagem adotada na aplicação do método e será abordada em detalhes no Capítulo 5. Porém, para contextualizar a interdependência entre diagramas, faz-se analogia a um exemplo de interdependência entre diagramas UML.

Em (Straeten et al, 2003), foi descrito um tipo de interdependência entre diagramas UML de classe e de sequência (OMG, 2004). Esta interdependência foi definida no contexto de evolução de software e se refere as classes que não

foram definidas em diagramas de classe, ou foram apagadas de diagramas de classe, mas cujas instâncias são utilizadas em diagramas de sequência. Neste caso a propriedade que está sendo violada é que toda instância utilizada num diagrama de sequência deve ser instância de uma classe definida em algum diagrama de classes pertencente à modelagem do sistema. Os autores deram o nome de *classless instance* à violação a esta propriedade. Consultas deste tipo são definidas, no Capítulo 5, para a linguagem de modelagem MAS-ML e no Capítulo 7 para a linguagem de modelagem AUML.

A terceira etapa da fase F2 é composta pelas consultas (QD2) que retornam como resposta sugestões de boas práticas de modelagem considerando-se todos os diagramas que compõem a modelagem de um SMA. Estas consultas são definidas para analisar a base de conhecimento que descreve a modelagem de um SMA e sugerir eventuais alterações que não se destinam a corrigir erros de modelagem, mas que podem melhorar sua qualidade, tornando-a, por exemplo, mais inteligível. É esperado que esta etapa seja enriquecida ao longo do tempo, com a definição de novas consultas, a partir do amadurecimento das metodologias e das linguagens de modelagem para este tipo de sistemas.

A fase F2 do Observed-MAS deve ser executada após serem feitas todas as correções solicitadas durante a fase F1, ou seja, quando não forem mais encontradas violações as propriedades previstas nas consultas de QV1. Esta estratégia se justifica, pois a presença de alguma violação as propriedades do domínio de SMAs ou as propriedades internas aos diagramas tornará inconsistente a base de conhecimento gerada considerando-se a estruturação definida pela Ont2, impedindo a utilização de aparato dedutivo na manipulação da base de conhecimento para análise das propriedades inter-diagramas.

4.5.

Discussão

Neste capítulo foi apresentada uma descrição formal para o arcabouço conceitual TAO através da definição de ontologias. As ontologias foram descritas em DL e tiveram sua consistência verificada pelo sistema RACER. O TAO é um arcabouço conceitual que define uma ontologia para o domínio de SMAs, e cujas entidades podem ser usadas como abstrações para definição de linguagens de modelagem e metodologias de desenvolvimento de SMAs. A partir da descrição

formal do TAO foi apresentada uma aplicação das ontologias na definição de um método para estruturação e análise de modelos de SMAs, o Observed-MAS.

O Observed-MAS é um método baseado na integração das ontologias usadas para descrever o TAO com a ontologia que descreve o metamodelo de uma linguagem de modelagem para SMAs. O método foi concebido para dar suporte à análise de modelos durante a etapa de projeto de SMAs, a fim de diminuir a incidência de erros durante as etapas mais avançadas de desenvolvimento dos sistemas. Esta análise é feita automaticamente a partir da transformação dos modelos de SMAs em instâncias da ontologia, armazenadas em bases de conhecimento que, por sua vez, são consultadas e manipuladas usando-se sistemas que possuem o aparato dedutivo necessário para manipular este tipo de dado.

A principal contribuição do capítulo foi a descrição formal do TAO e verificação de sua consistência. Durante esta descrição foram observadas algumas lacunas na especificação das entidades organizações e ambientes ativos, pois não estava explícito no TAO como tais entidades interagem com os agentes e as sub-organizações do sistema. Para tanto, foram introduzidos protocolos dentre os elementos que descrevem as propriedades internas de organizações e ambientes ativos, ou seja, das entidades proativas que não desempenham papéis. O método proposto como aplicação das ontologias também é uma contribuição, que terá sua extensão avaliada ao longo dos capítulos seguintes, quando é feita a aplicação do método para duas linguagens de modelagem orientada a agentes, com respectivos estudos de caso.

A análise de quão adequada é a formalização do TAO como ontologia que descreve o domínio de SMAs, e que pode ser usada como base para prover suporte à análise automática de propriedades de SMAs abordadas por métodos, metodologias, linguagens de modelagem e padrões de arquitetura que auxiliem o desenvolvimento de SMAs, é uma questão que depende da realização de vários estudos empíricos.